Performance Analysis of Distributed Intrusion Detection Protocols for Mobile Group Communication Systems

Jin-Hee Cho
Computational & Information Sciences Directorate
U.S. Army Research Laboratory
jinhee.cho@us.army.mil

Ing-Ray Chen
Department of Computer Science
Virginia Tech
irchen@vt.edu

Abstract

Under highly security vulnerable, resource restricted, and dynamically changing mobile ad hoc environments, it is critical to be able to maximize the system lifetime while bounding the communication response time for mission-oriented mobile groups. In this paper, we analyze the tradeoff of security versus performance for distributed intrusion detection protocols employed in mobile group communication systems (GCSs). We investigate a distributed voting-based intrusion detection protocol for GCSs in multi-hop mobile ad hoc networks and examine the effect of intrusion detection on system survivability measured by the mean time to security failure (MTTSF) metric and efficiency measured by the communication cost metric. We identify optimal design settings under which the MTTSF metric can be best traded off for the communication cost metric or vice versa.

1. Introduction

Developing network security protocols in mobile ad hoc networks (MANETs) has design challenges due to high security vulnerabilities and unique characteristics of MANET environments such as open medium, dynamic changing network topology, decentralized decision-making and cooperation, lack of centralized authority, lack of resources in mobile devices (e.g., bandwidth, memory, computational power), and no clear line of defense [7]. Under these environmental constraints, intrusion detection mechanisms are used as a second line of defense and have become essential for systems with the goal of high-survivability in the presence of inside attackers.

This paper concerns distributed intrusion detection employed in mission-oriented group communication systems (GCSs) in MANETs for detecting and evicting compromised nodes. Examples of mission-oriented GCSs include rescue teams with mobile devices in disaster management, soldiers with mobile devices in battlefield situations, mobile robots with embedded sensor systems to explore the surface of Mars [11], and mobile tanks with sensors to survey a hostile battlefield for tracking bio/chemical plumes [11]. In these mission-oriented mobile applications for MANETs, guaranteeing maximum system lifetime while minimizing bandwidth consumed is critical for successful mission execution.

Many emerging mobile applications depend on the notion of secure group communication where mobile nodes can join or leave a group dynamically [9]. A compromised node in a group can compromise the security of the system when useful information has been leaked out to compromised nodes. Compromised nodes may also collude. To tolerate/detect intrusions, it is essential to dynamically detect and evict compromised nodes by adaptively enhancing the defense in response to the attacker strength. On the other hand, if IDS is performed unnecessarily frequent, it may adversely affect the performance of GCS.

While intrusion detection systems (IDS) for wired networks have been extensively studied, there has been little work on IDS for wireless mobile environments, particularly for MANETs. Zhang *et al*. [15] pioneered a distributed and cooperative intrusion detection model based on anomaly detection by which all nodes in the system run IDS to detect and respond to intrusion. Nevertheless, no specific IDS or reactive IDS protocol was discussed. Cluster-based IDS for MANETs has been proposed [1, 12, 14]. The main idea is that a cluster head (CH) collects security-related information from nodes in the same cluster and determines if any intrusion has occurred. An important issue not addressed is performance degradation due to zone-based IDS. Marti *et al*. [6] developed a *watchdog* mechanism for identifying misbehaving nodes based on dynamical behaviors and developed a *pathrater* algorithm for routing around misbehaving nodes for MANETs. Hierarchical IDS was proposed to realize distributed anomaly-based IDS in MANETs [1, 4]. However, the issues of extra latency and energy consumption were not addressed. [1, 4] only examined security properties of IDS in MANETs without dealing with reactive approaches against any changes of attackers' behaviors. Stern *et al*. [12] proposed data reduction techniques to reduce communication costs in their IDS design. However, detection latency introduced by data aggregation and their effect on performance degradation were not investigated.

Our work differs from prior work in that we consider the impact of IDS on performance degradation, and identify optimal design settings of adaptive IDS based on the

tradeoff between performance (i.e., communication cost) versus security (i.e., mean time to security failure or MTTSF) for GCSs in MANETs.

Recently, Subhadrabandhu *et al*. [13] studied the tradeoff between energy/computational/communication resource consumption versus IDS accuracy based on distributed IDS. Algorithms were developed to explore the tradeoff. Our work differs from their work in that we specifically deal with GCSs in MANETs as well as propose a framework to identify optimal design settings of adaptive distributed IDS under system-imposed security and performance requirements. To evaluate both security and performance characteristics of IDS in GCSs, the approach used in our work has its root in model-based quantitative analysis [8]. In the literature, we have seen some recent work extending model-based quantitative analysis to security analysis. Madan *et al*. [5] employed a Semi-Markov Process (SMP) model to evaluate security attributes of an intrusion tolerant system to obtain dependability measures such as availability, and a transient analysis with absorbing states to obtain security measures such as *mean time to security failure* (MTTSF). Most of the previous work cited above, however, often only focused on security measures without considering the impact of deploying security mechanisms on the performance of the system. Further, they did not deal with system optimization issues in terms of identifying optimal design settings of adaptive IDS based on the tradeoff for mission-oriented mobile groups.

The contributions of the paper are as follows. First, we develop quantitative analysis methods to analyze the tradeoff between security and performance of mobile GCSs in MANETs, in the presence of inside attackers and intrusion detection mechanisms, recognizing the fact that security mechanisms often have impacts on the performance property of the system. We develop an analytical model based on Stochastic Petri nets (SPN) to succinctly describe the inside attacker, the GCS, and distributed intrusion detection mechanisms with the goal to evaluate the effect of intrusion detection on security and performance properties of the system. Second, when given a set of parameter values characterizing the operational or environmental conditions of the system, we identify the optimal intrusion detection interval under which the MTTSF metric is maximized. Moreover, we effectively trade security off for performance, or vice versa, such that system designers can adjust the intrusion detection interval to maximize MTTSF while satisfying imposed performance requirements in terms of overall communication cost. Third, we propose a robust, efficient, and adaptive distributed intrusion detection mechanism that dynamically adjusts the intrusion detection interval and a detection function optimally reacting to dynamically changing attacker strength. Our IDS protocol considers the effect of possible collusion of compromised nodes as well as IDS intrinsic defects including false positives and false negatives. Lastly, we model our secure GCS for MANETs based on realistic behaviors of inside attackers.

2. Preliminary

2.1 Secure Group Communication Systems in Mobile Ad Hoc Networks

Secure GCSs are most often seen in military settings where combat units spread out in a geographical area without a communication infrastructure but must maintain a consistent view in order to make correct combat decisions. Group members must communicate each other state changes, such as changes of membership of nodes, their locations, and approaching objects. Very typically, such a military deployment is mission-oriented and the goal is to complete the combat mission within its system lifetime. In this sense, the security requirement is expressed in terms of a threshold for MTTSF such that the system must be able to survive security threats past the minimum mission time. The timeliness requirement is the delay requirement per packet. This translates into a maximum network traffic rate which bounds the delay or response time per packet.

An efficient way to achieve secure group communications is to use a symmetric key, called the *group key*, shared by group members. Group members may agree upon the group key by means of a group key agreement protocol in a MANET in which there is no centralized key server. Group members employ the group key to encrypt group messages. By employing the group key as a secret key, only members of the group are able to decrypt and read group messages [9]. This achieves the *confidentiality* property for secure group communications.

In a dynamic group setting where users can join or leave the group at any time, the group key needs to be rekeyed. There are the two main security properties commonly associated with rekeying [9], namely, *forward secrecy*, which means that an adversary who knows previous group keys cannot identify subsequent group keys, and *backward secrecy*, which means that an adversary who knows the current group key cannot discover previous group keys. To maintain both backward and forward secrecy, rekeying (i.e., change the group key) is performed whenever a group membership change event occurs due to a user newly joining the group or a current member leaving or being evicted.

For MANETs, there is no centralized trusted key server. Instead a distributed key management protocol, a *contributory key agreement* (CKA) protocol, would be used for rekeying upon a join/leave/eviction event. Without loss of generality, this paper uses a CKA protocol with GDH [10] for secret key generation in a distributed way.

2.2 Distributed Intrusion Detection Protocols

We consider two types of intrusion detection protocols applicable to GCSs in MANETs, i.e., *host-based IDS* versus *voting-based IDS*. The first type is ***host-based IDS*** [15] in which each node performs local detection to determine if a neighboring node has been compromised. Each node may preinstall host-based IDS with standard existing IDS techniques such as misuse detection (also called signature-based detection) and anomaly detection [15] so

that our proposed voting-based IDS can be independent of the host-based IDS technique used as a general framework. Each node may evaluate its neighbors based on information collected, mostly route-related and traffic-related information [4]. We measure the effectiveness of IDS techniques applied (e.g., misuse detection or anomaly detection) for host-based IDS by two parameters, namely, the false negative probability (*p1*) and false positive probability (*p2*). In general, when the system uses misuse detection for IDS, it tends to have more false negatives and less false positives (e.g., higher *p1* and lower *p2*). On the other hand, when the system employs anomaly detection for IDS, it is likely to have fewer false negatives and more false positives (e.g., lower *p1* and higher *p2*).

The second type is ***voting-based IDS*** for cooperative detection based on majority voting. Voting-based IDS derives from the fault tolerance concept based on majority voting for evicting a target node in the context of sensor networks [2]. For voting-based IDS to be performed, each node again is preinstalled with host-based IDS to collect information to detect the status of neighboring nodes. Periodically a target node would be evaluated by *m* vote-participants dynamically selected where *m* is a design parameter. If the majority of *m* nodes decided to vote against the target node, then the target node would be evicted from the system by means of rekeying. This adds intrusion tolerance to tolerate collusion of compromised nodes in MANETs as it takes the majority of "bad" nodes among *m* nodes to work against the system. We characterize voting-based IDS by two parameters, namely, false negative probability (P_{fn}) and false positive probability (P_{fp}). These two parameters can be calculated based on (a) the *per-node* false negative and positive probabilities (*p1* and *p2*) of host-based IDS in each node; (b) the number of vote-participants, *m*, selected to vote for or against a target node; and (c) an estimate of the current number of compromised nodes which may collude with the objective to disrupt the service of the system. Since *m* nodes are selected to vote, if the majority of *m* voting-participants (i.e., $\geq \lceil m/2 \rceil$) cast negative votes against a target node, the target node is regarded as compromised and will be evicted from the system.

3. System Model

This paper concerns a mission-oriented GCS consisting of mobile groups in MANETs equipped with intrusion detection to mainly deal with inside attackers. We define a mobile group based on "connectivity." When all nodes are connected, there is only a single group in the system. That is, group members must maintain connectivity for them to be in the same group. The GCS and its constituent mobile groups are "mission-oriented" in the sense that a mobile group may be partitioned into several groups due to network partition derived from node mobility or node failure, but each group will continue to execute the mission amid group partition and merge activities. Moreover, the GCS fails when any mobile group fails, modeling the case in which a security failure of any mobile group compromises the mission assigned, e.g., in secret mission situations. We assume that each member has a private key and its certified public key available for *authentication* purposes. When a new member joins a mobile group, the new member's identity is authenticated based on the member public/private key pair by applying the challenge/response mechanism. We assume that the GCS maintains *view synchrony (VS)* [9] by which messages are guaranteed to be delivered reliably and in order. We characterize the workload and operational conditions of a GCS in MANETs by a set of model parameters. We assume that the inter-arrival times of join and leave requests are exponentially distributed with their rates being λ and μ respectively. Also we assume that the inter-arrival time of data packets issued by a node for group communication is exponentially distributed with rate λ_q. The assumption of exponential distribution can be relaxed since the SPN performance model developed is capable of allowing any general distribution for a transition time. We assume that the time to perform a rekeying operation upon a membership change event (i.e., join or leave event) or a forced eviction is measured based on GDH [10] to realize distributed key management in MANETs.

Recognizing the principles [3] that attacker behaviors are not always random, we use three attacker functions to model the attacker strength based on the prediction of time and effort made to perform an attack as follows:

- ***Logarithmic time attacker***: The attacker increasingly takes longer time to compromise nodes in the system, following a logarithmic function curve.
- ***Linear time attacker***: The attacker compromises nodes one after the other with the node compromising rate linear to the number of compromised nodes in the system.
- ***Polynomial time attacker***: The attacker increasingly takes shorter time to compromise nodes in the system, following an exponential function curve.

We assume that IDS will perform its function periodically. The detection interval is dynamically adjusted in response to the accumulated number of intrusions that have been detected in the system. Similar to the attacker behavior model above, we consider three detection functions to model the IDS activities in terms of periodicity, namely, *logarithmic*, *linear*, and *polynomial periodic detection*, as follows:

- ***Logarithmic periodic detection***: In this detection scheme, the system performs intrusion detection in a conservative way with a rate logarithmic to the number of compromised nodes that have been identified.
- ***Linear periodic detection***: This system performs IDS with a linear rate to the number of intrusions that have been detected in the system.
- ***Polynomial periodic detection***: This detection scheme aggressively performs IDS by increasing the detection rate in a polynomial fashion to the number of observed compromised nodes in the system.

To alleviate collusion among compromised nodes, the system performs voting-based IDS by which the majority of vote-participants must agree to evict a target node before the target node is evicted. The number of vote-participants (*m*)

is a system parameter whose effect will be analyzed in the paper.

We define two security group failure conditions so that a mobile group enters a security failure state when one of the two security group failure conditions stated below is true. That is, the GCS fails if any of mobile groups fails when either Condition C1 or C2 listed below is true.

- **Condition C1**: a compromised but undetected member requests and subsequently obtains data using the group key. The system is in a failure state because data have been leaked out to a compromised node, leading the *loss of system integrity* in a security sense.
- **Condition C2**: more than 1/3 of member nodes are compromised but undetected by IDS. We assume the *Byzantine Failure* model [3] such that when more than 1/3 of member nodes in a mobile group are compromised, the mobile group is compromised, resulting in the *loss of availability* of system services.

If a member node is detected as compromised by IDS, the system won't allow the member node to request data anymore and will evict the member immediately to satisfy the forward/backward secrecy requirement. After a node is detected as compromised and evicted from the system, it cannot rejoin the group again. That is, there is no recovery mechanism available in the system to repair a compromised member and make it a trusted member node again. Initially, all nodes are trusted.

We assume that there are intrusion prevention techniques in place, such as encryption or authentication, to deal with outsider attacks (e.g., disrupting traffic, modifying data, eavesdropping). Insider attacks are due to compromised nodes disguised as legitimate members to disrupt the system. We also consider the behaviors of inside attackers such as obtaining secret information and accordingly passing it to outside attackers (i.e., illegal data leak out) or disseminating a fake vote to keep more compromised nodes but evict good nodes from the system.

3.1 Security and Performance Metrics

- ***MTTSF (Mean Time to Security Failure)***: This metric indicates the average time elapsed for the GCS to reach a security failure state. The GCS fails when any mobile group reaches a security failure state when (1) data have been leaked out to a compromised but undetected member node (i.e., C1), or (2) more than *1/3* of the member nodes have been compromised (i.e., C2). Note that illegal data leak out only occurs when a compromised but undetected member requests data and subsequently obtains data using the group key. As a security metric, lower MTTSF means faster *loss of system integrity* or *loss of availability*. A design goal is to maximize MTTSF.
- ***Communication Traffic Cost ($\hat{C}_{total}$):*** This metric indicates the total traffic incurred per time unit (*s*) including group communication, status exchange, rekeying, intrusion detection, beacon, group partition/merge and mobility-induced activities. Since all nodes share the wireless bandwidth (*BW*), a high $\hat{C}_{total}$ will be translated into a high level of contention and consequently a high delay or response time for group communication. A design goal is to minimize $\hat{C}_{total}$.

4. Performance Model

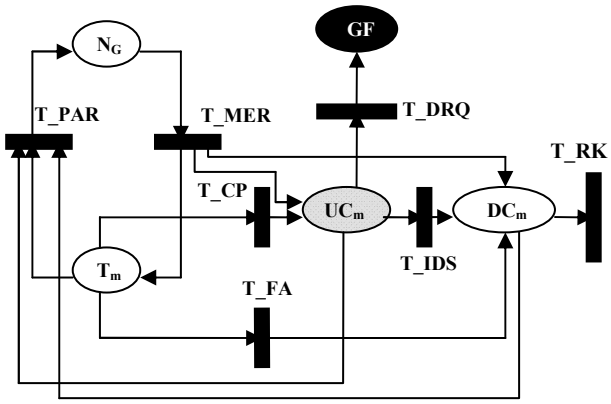

Figure 1: SPN Model.

We develop a SPN model as shown in Figure 1 to describe the behavior of a mobile group in the presence of insider attacks and intrusion detection activities, with the goal of identifying optimal design settings (i.e., optimal intrusion detection interval and detection function) to maximize MTTSF while satisfying imposed performance (i.e., overall communication cost) requirements. Here we describe how the SPN model is constructed as follows:

- The SPN model describes the behavior of a *single* mobile group as it evolves. This mobile group may partition into two and may merge with another group during its lifetime. We track trusted members, compromised members undetected, and compromised members detected during the group's lifetime to understand its security and performance characteristics. Certainly the system knows the number of compromised nodes detected by IDS at all times. However, the system does not know compromised nodes not yet detected. It only knows the total number of member nodes. The SPN model predicts the number of compromised but yet detected nodes through knowledge of the node compromising rate or the attacker function explained below.
- We use places to classify nodes except for place N_G which holds the current number of groups in the system. Specifically, place T_m holds trusted members, UC_m holds compromised nodes not yet detected by IDS, and DC_m holds compromised nodes that have been detected by IDS. Note that T_m, UC_m, and DC_m represent nodes in one group, not in the system. To be more specific, the numbers of nodes in places T_m, UC_m, and DC_m, obtained by $mark(T_m)$, $mark(UC_m)$, and $mark(DC_m)$, respectively, would be adjusted based on the number of groups existing in the system (obtained by $mark(N_G)$), which changes upon group merge/partition events.
- We use transitions to model events. Specifically, *T_MER* and *T_PAR* model the group merge or partition events, respectively; *T_CP* models a node being compromised. *T_FA* models a node being falsely identified as compromised. *T_IDS* models a compromised node being

detected. *T_RK* models rekeying. *T_DRQ* models a data leak security failure (i.e., C1). A firing of a transition will change the state of the system, which is represented by the distribution of tokens in the SPN. For example, $mark(N_G)$ changes upon firing *T_MER* or *T_PAR* since the number of groups changes upon a group merge or partition event; the number of compromised nodes undetected increments by 1 and, place UC_m will hold one more token when *T_CP* fires. A transition is eligible to fire when the firing conditions associated with the event are met. The firing conditions are (1) its input place must contain at least one token and (2) the associated enabling guard function, if exists, must return *true*. For example, *T_CP* is enabled to fire when there exists "good" nodes in the group, that is, place T_m holds at least one token, and the enabling function associated with *T_CP* returns *true*.

- Except for tokens contained in place N_G, we use a "token" in the SPN model to represent a node in the group. The population of each type of nodes is equal to the number of tokens in the corresponding place. Initially, all *N* members are trusted in one group and put in place T_m as tokens.
- Trusted members may become compromised because of insider attacks with a node-compromising rate $A(m_c)$. This is modeled by firing transition *T_CP* and moving tokens one at a time (if it exists) from place T_m to place UC_m.
- Tokens in place UC_m represent compromised but undetected member nodes. We consider the system as having experienced a security failure when data are leaked out to compromised but undetected members, i.e., C1. A compromised and undetected member will attempt to compromise data from other members in the group. Because of the use of host-based IDS, a node will reply to such a request only if it could not identify the requesting node as compromised with the false negative probability *p1*. This is modeled by associating transition *T_DRQ* with rate $p1*\lambda_q*mark(UC_m)$. The firing of transition *T_DRQ* will move a token into place *GF*, at which point we regard the system as experiencing a security failure due to C1.
- A compromised node in place UC_m may be detected by IDS before it compromises data in the GCS. The intrusion detection activity of the mobile group is modeled by the IDS detection rate $D(m_d)$. Whether the damage has been done by a compromised node before the compromised node is detected depends on the relative magnitude of the node-compromising rate ($A(m_c)$) versus the IDS detection rate ($D(m_d)$). When transition *T_IDS* fires, a token in place UC_m will be moved to place DC_m, meaning that a compromised but undetected node now becomes detected by IDS. For voting-based IDS, the transition rate of *T_IDS* is $mark(UC_m)*D(m_d)*(1-P_{fn})$, taking into consideration of the number of compromised but yet detected nodes and the false negative probability of voting-based IDS. Voting-based IDS can also false-positively identify a trusted member node as compromised. This is modeled by moving a trusted member in place T_m to place DC_m after transition *T_FA* fires with rate $mark(T_m)*D(m_d)*P_{fp}$. Note that voting-based IDS parameters, P_{fn} and P_{fp}, can be derived based on *p1* and *p2*, the number of vote-participants (*m*), and the current number of compromised nodes which may collude to disrupt the service of the system. Later we will show how we may parameterize P_{fn} and P_{fp}.
- A mobile group experiences a security failure if either security failure condition, C1 or C2, is met. We model this by making the group enter an absorbing state when either C1 or C2 is *true*. To achieve this, we associate every transition in the SPN model with an enabling function that returns *false* (disabling the transition from firing) when either C1 or C2 is met, and returns *true* otherwise. For the SPN model, C1 is *true* when $mark(GF) > 0$ representing that data have been leaked out to compromised, undetected members; C2 is *true* when more than 1/3 of member nodes are compromised but undetected as indicated by $mark(UC_m)/(mark(T_m) + mark(UC_m)) > 1/3$.

4.1 Parameterization

Here we describe the parameterization process, i.e., how to give model parameters proper values reflecting the operational and environmental conditions of the system.

- **T_{cm}:** Recall that T_{cm} is the communication time required for broadcasting a rekey message for a join or leave event based in GDH. The reciprocal of T_{cm} is the rate of transition *T_RK*.
- **$A(m_c)$:** This is an attacker function that returns the rate at which nodes are compromised in the mobile group. It will apply to transition *T_CP* in the SPN model. Three different attacker strengths are considered based on the time taken to compromise a node, namely, *logarithmic*, *linear*, and *polynomial time* given by $A_{log}(m_c) = \lambda_c log_p(m_c)$, $A_{linear}(m_c) = \lambda_c m_c$, and $A_{poly}(m_c) = \lambda_c (m_c)^p$ respectively, where $m_c = (mark(T_m) + mark(UC_m)) / mark(T_m)$. These three attacker strengths differ by the way the node compromising rate increases as more nodes become compromised. For the linear attacker function, the node compromised rate increases linearly with the number of compromised nodes. Hence, $A_{linear}(m_c) = \lambda_c m_c$ where m_c reflects the degree of compromised nodes currently in the group and λ_c is the base node compromising rate initially given that there is no compromised node in the group. For $A_{log}(m_c)$, the compromising rate increases in a logarithm form with the number of compromised nodes. For $A_{poly}(m_c)$ the compromising rate increases in an exponential form with the number of compromised nodes. Note that these three forms are prediction functions for the node compromising rate. The base compromising rate (λ_c) can be obtained by first-order approximation from observing the number of compromised nodes over a time period. We also note that *p* is a base index parameter selected to reflect the degree of changes of the logarithmic and polynomial attacker functions with respect to the number of compromised nodes. It requires fine tuning after sufficient data are collected. We choose p=3 in this paper.
- **$D(m_d)$:** This is a detection function that returns the rate at which IDS is invoked. Three different detection functions, namely, *logarithmic*, *linear*, and *polynomial periodic detection,* are given by $D_{log}(m_d) = log_p(m_d)//T_{IDS}$, $D_{linear}(m_d) = m_d/T_{IDS}$, and $D_{poly}(m_d) = (m_d)^p/T_{IDS}$, respectively, where m_d

$= N_{init}/(mark(T_m) + mark(UC_m))$. These three functions differ by the way the detection rate changes with the number of compromised nodes that have been detected by IDS. For the linear detection function, the IDS detection rate increases linearly with the number of compromised nodes detected. D_{linear} (m_c) is the linear periodic detection function where m_c indicates the degree of compromised nodes that have been detected by IDS, and T_{IDS} is the base detection time interval which we aim to determine for maximizing MTTSF when applying voting-based IDS. The log detection function, D_{log} (m_d), and exponential detection function, D_{poly} (m_d), have the same form as their counterparts in the attacker function. We note again that p is a base index parameter selected to reflect the degree of changes of the logarithmic and polynomial detection functions with respect to the number of compromised nodes detected. We again choose p=3 in this paper.

- **Group Merge and Partition:** We model group merge and partition events by a *birth-death process* with birth modeling group partitioning and death modeling group merging. We obtain *group* merging/partitioning rates by simulation for a sufficiently long period of time.
- $\boldsymbol{P_{fn}}$, $\boldsymbol{P_{fp}}$: P_{fn} is the probability of false negatives defined as the number of compromised nodes diagnosed by voting-based IDS as trusted healthy nodes (i.e., detecting a bad node as a good node) over the number of detected nodes. On the other hand, P_{fp} is the probability of false positives defined as the number of normal nodes flagged as anomaly over the number of trusted normal nodes. We consider the intrinsic defect of host-based IDS in each node as well as the possible collusion of compromised nodes during the voting process. If a vote-participant is compromised, it can cast a negative vote to evict a healthy target node in the group or it can cast a positive vote for a malicious node to keep more compromised nodes in the group. Equation 1 reflects these two cases of false positives or false negatives introduced into the group respectively, where $N_{majority} = \lceil m/2 \rceil$ and $N = mark(T_m) + mark(UC_m)$. Here m is the number of vote-participants to cast a vote against a target node, p_f is $p1$ for calculating P_{fn} *or* $p2$ for calculating $P_{fp,}$ N_{bad} is the number of currently compromised nodes in the group represented as $mark(UC_m)$, and N_{good} is the number of currently healthy nodes in the group indicated as $mark(T_m)$. P_{fp} is obtained when the majority of voters consists of bad nodes who cast a negative vote against a good node, and good nodes who mistakenly diagnose a good node as a bad node with the probability of $p2$ (i.e., $p2$ is a per-node false positive probability), resulting in a healthy node being evicted. On the other hand, P_{fn} occurs when the majority of voters is from positive votes by bad nodes (i.e., casting a positive vote against a bad node) or good nodes who mistakenly diagnose a bad node as a good node with the probability of $p1$ (i.e., $p1$ is a per-node false negative probability), keeping more compromised nodes undetected in the group. Note that P_{fn} and P_{fp} are constantly being adjusted to properly react to dynamically changing network and operational conditions, such as the degree of compromised nodes, node density, and number of vote-participants (m) used over time.

$$P_{fp}\ or\ P_{fn} = \sum_{i=0}^{m-N_{majority}} \left[\frac{C\left(\binom{N_{bad}}{N_{majority}+i} \times C\left(\binom{N_{good}}{m-(N_{majority}+i)}\right)\right)}{C\binom{N_{good}+N_{bad}}{m}} \right] + \sum_{i=0}^{m-N_{majority}} \left[\frac{C\binom{N_{bad}}{i} \times \sum_{j=N_{majority}-i}^{m-i} \left[C\binom{N_{good}}{j} \times p_f{}^{j} \times C\binom{N_{good}-j}{m-i-j} \times (1-p_f)^{(m-i-j)} \right]}{C\binom{N_{good}+N_{bad}}{m}} \right] \quad (1)$$

4.2 MTTSF and $\hat{C}_{total}$ Calculation

MTTSF is obtained using the concept of *mean time to absorption* (*MTTA*) in the SPN model. Specifically, we use a reward assignment such that a reward of 1 is assigned to all states except for absorbing states in which Condition C1 or C2 is met. The MTTSF of the system is simply the expected accumulated reward until absorption, defined as $E[Y(\infty)] = \sum_{i \in S}^{\infty} r_i \int_0^{\infty} P_i(t)dt$ where S denotes the set of all states except absorbing states, r_i (reward) is 1, and $P_i(t)$ is the probability of state i at time t.

We calculate $\hat{\mathbf{C}}_{\mathbf{total}}$ by the probability-weighted average of $\hat{C}_{total,i}$ representing the communication cost incurred per time unit (s) in state i. Specifically, $\hat{C}_{total}$ is calculated by accumulating $\hat{C}_{total,i}$ (t) over MTTSF divided by MTTSF, i.e., $\hat{C}_{total} = \int_0^{MTTSF} \hat{C}_{total,i}(t)dt/MTTSF$, where $\hat{C}_{total,i} = \hat{C}_{GC,i} + \hat{C}_{status,i} + \hat{C}_{rekey,i} + \hat{C}_{IDS,i} + \hat{C}_{beacon,i} + \hat{C}_{mp,i}$ and $\hat{C}_{GC,i}$, $\hat{C}_{status,i}$, $\hat{C}_{rekey,i}$, $\hat{C}_{IDS,i}$, $\hat{C}_{beacon,i}$, *and* $\hat{C}_{mp,i}$ are costs per time unit for group communication, status exchange, rekeying, intrusion detection, beacon, group partition and group merge, and mobility events, respectively, given that the number of groups in the system is i. Due to space limitation, we omit calculation steps here.

5. Numerical Data and Analysis

We present numerical data obtained through the evaluation of the SPN model developed and provide physical interpretations. Our objective is to identify optimal intrusion detection interval (T_{IDS}) that will maximize MTTSF while satisfying performance requirements of the system. We also aim to identify the best detection function to use in response to the attacker function (compromising rate) detected at runtime. We vary the values of key parameters including the number of vote-participants in voting-based IDS (m=5 as the default), group communication rate (λ_q=1 per minute), base compromising rate (λ_c=1 per 12 hrs), and the attacker and detection functions to analyze their effects on the optimal base detection interval for maximizing MTTSF. The default values used for other parameters include the operational area of the MANET environment (radius=500 m), the number of nodes (N=100), per-node join rate (λ=1 per hr),

per-node leave rate (μ=1 per 4 hrs), the wireless bandwidth (*BW*=1Mbps), and host-based IDS false positive or false negative probabilities (*p1* = *p2* = 1% since in general 1% or less is considered acceptable). Here we note that *p1* and *p2* are two probability parameters for characterizing any host-based IDS preinstalled in each node. Each node moves according to the random waypoint mobility model.

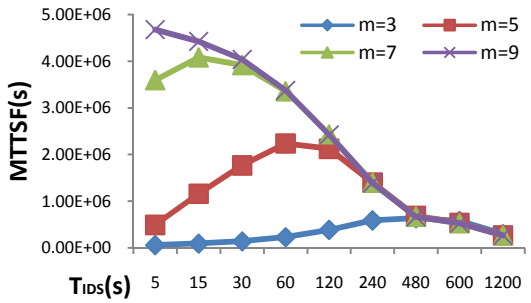

Figure 2: Effect of *m* on MTTSF and Optimal T_{IDS}.

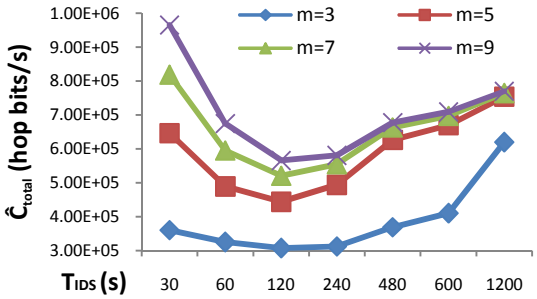

Figure 3: Effect of *m* on $\hat{C}_{total}$ and Optimal T_{IDS}.

We first analyze the effect of intrusion detection interval (T_{IDS}) on MTTSF as a function of the number of vote-participants (m) and demonstrate that there exists an optimal intrusion detection interval (T_{IDS}) for maximizing MTTSF or minimizing $\hat{C}_{total}$. Figure 2 shows the effect of intrusion detection interval (T_{IDS}) on MTTSF as the number of vote-participants (m) changes for the case in which the attacker function and the detection function are both linear. We observe that there exists an optimal T_{IDS} that maximizes MTTSF for each given m value. In general, as T_{IDS} becomes larger, MTTSF increases until its optimal point reaches, and then MTTSF decreases after the optimal point. The reason of increasing MTTSF as T_{IDS} increases initially is that as T_{IDS} increases there are fewer nodes being falsely identified by IDS since IDS is triggered less often, thus reducing the system failure probability due to C2. After the optimal T_{IDS} is reached, MTTSF decreases again because IDS is not triggered often enough to detect compromised nodes which may perform attacks to cause system failures due to C1. Note that P_{fp} is one aspect of false alarms generated by IDS, and therefore more nodes will be falsely identified as compromised nodes if IDS is more frequently triggered.

From Figure 2 we also observe the sensitivity of optimal T_{IDS} identified on MTTSF as m varies. When m is large, the false alarm probability (P_{fp} + P_{fn}) is small because more nodes are participating in the voting process, reducing the possibility of collusion by compromised nodes. Consequently, when m is large, we observe a high MTTSF due to the small false alarm probability. Conversely, when m is small, MTTSF is small due to a larger false alarm probability. A smaller m also results in a longer optimal T_{IDS} being used to maximize MTTSF to offset the adverse effect of IDS with large false positives, e.g., optimal T_{IDS} = 480, 60, 15, and 5 *s* for m = 3, 5, 7, and 9 respectively.

Figure 3 shows the overall communication cost ($\hat{C}_{total}$) versus the intrusion detection interval (T_{IDS}) as the number of vote-participants (m) varies. An optimal T_{IDS} exists in each curve (for minimizing $\hat{C}_{total}$) because of the tradeoff between decreasing normal group communication costs ($\hat{C}_{GC,i}$) and increasing IDS related communication costs ($\hat{C}_{eviction,i}$ + $\hat{C}_{IDS,i}$) as T_{IDS} becomes shorter. Also we observe that when m is large, $\hat{C}_{total}$ is high. This is because a larger m induces a lower P_{fp} under which more nodes will be able to perform normal group activities. Furthermore, when there are more vote participants, there is a higher cost associated with dynamic majority voting. Contrary to MTTSF versus T_{IDS}, we do not observe the sensitivity of an optimal T_{IDS} identified, but there is a relatively higher communication cost saved when the optimal T_{IDS} identified is employed as m increases.

Next we analyze how one can select the best detection interval (T_{IDS}) and detection function D (m_d) to optimize MTTSF while satisfying the performance requirement in terms of communication overhead, when given the attacker function A (m_c) detected at runtime. In Figure 4, we show MTTSF versus T_{IDS} for the three detection functions $D(m_d)$ given that the attacker function is linear. We see that each curve again has its own optimal T_{IDS}. The linear detection function $D_{linear}(m_d)$ shows the best performance at T_{IDS} = 120 s generating the highest MTTSF overall, while the logarithmic detection function $D_{log}(m_d)$ is the worst, particularly when T_{IDS} is sufficiently small. This tradeoff is attributed to the speed of detection (log, linear, or exponential) versus the speed of attack (linear). If the former is greater than the latter, many false positives may be generated; conversely, many compromised nodes may remain in the system. The linear detection function that matches up with the linear attacker function is the best among the three detection functions in terms of the tradeoff of the two ends. With similar reasoning, we see that the strongest polynomial detection function $D_{poly}(m_d)$ performs well when T_{IDS} is large (e.g., T_{IDS} > 240 s) while the weakest logarithmic detection function $D_{log}(m_d)$ performs well when T_{IDS} is small (T_{IDS} < 15 s).

Correspondingly Figure 5 shows the overall communication cost ($\hat{C}_{total}$) versus T_{IDS} for the three detection functions $D(m_d)$ given that the attacker function is linear. Each curve in Figure 5 also has an optimal T_{IDS} that minimizes $\hat{C}_{total}$. The general trend of the optimal T_{IDS} identified is similar to that shown in Figure 4 except that the exact optimal T_{IDS} points are different. The best performance of $\hat{C}_{total}$ is observed with linear detection at T_{IDS} = 240 s while the worst performance of $\hat{C}_{total}$ is shown with logarithmic detection under the ranges of T_{IDS} > 120 s and with polynomial detection under the ranges of T_{IDS} ≤ 120 s.

In terms of the optimal T_{IDS} identified to minimize $\hat{C}_{total}$, we see that a shorter optimal T_{IDS} is preferred with less aggressive logarithmic detection, since a shorter T_{IDS} contributes to nodes being evicted more often, consequently leading to less group communication activities. On the other hand, as the detection function becomes aggressive, i.e., polynomial detection, a longer optimal T_{IDS} is favorable to minimize $\hat{C}_{total}$ in order not to increase too much IDS related traffic more than needed due to aggressive IDS.

These results demonstrate that the system could adjust the IDS detection strength in response to the attacker strength detected at runtime in order to maximize MTTSF and minimize $\hat{C}_{total}$ dynamically. By selecting the best detection function (*logarithmic*, *linear*, or *polynomial*) in response to the attacker strength, we can maximize MTTSF without experiencing much of the adverse effect of IDS. The results obtained in terms of MTTSF and $\hat{C}_{total}$ versus T_{IDS} allow the system designer to select the best intrusion detection interval (T_{IDS}) to maximize MTTSF while satisfying the $\hat{C}_{total}$ performance requirement.

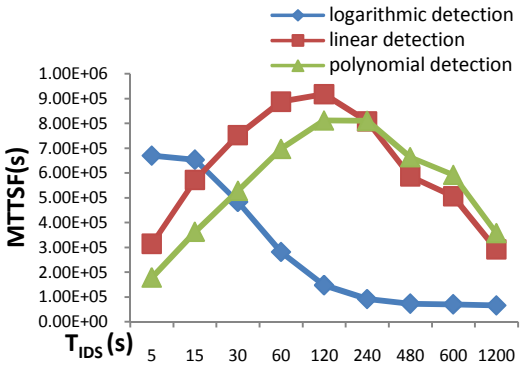

Figure 4: Effect of T_{IDS} on MTTSF with respect to $D(m_d)$ under linear time attackers when $m = 5$.

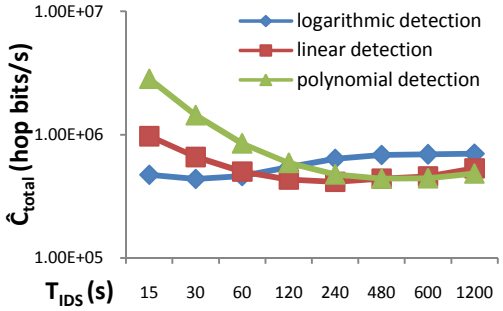

Figure 5: Effect of T_{IDS} on $\hat{C}_{total}$ with respect to $D(m_d)$ under linear time attackers when $m = 5$.

Acknowledgements

This project is supported in part by an appointment to the U.S. Army Research Laboratory Postdoctoral Fellowship Program administered by the Oak Ridge Associated Universities through a contract with the U.S. Army Research Laboratory.

References

[1] J. B. D. Cabrera, C. Gutierrez, R. K. Mehra, "Infrastructures and Algorithms for Distributed Anomaly-based Intrusion Detection in Mobile Ad-hoc Networks," *IEEE Military Comm. Conf.*, vol. 3, Oct. 2005, pp. 1831 – 1837.

[2] H. Chan, V. D. Gligor, A. Perrig, and G. Muralidharan, "On the Distribution and Revocation of Cryptographic Keys in Sensor Networks," *IEEE Trans. on Dependable and Secure Computing*, vol. 2, no 3, 2005, pp. 233 – 247.

[3] F. C. Gärtner, "Byzantine Failures and Security: Arbitrary is not (always) Random," *Technical Report IC/2003/20,* Swiss Federal Institute of Technology School of Computer and Communication Sciences, April, 2003.

[4] Y. A. Huang and W. Lee, "A Cooperative Intrusion Detection System for Ad Hoc Networks," *Proc. 1st ACM Workshop on Security of Ad-hoc and Sensor Networks*, Fairfax, Virginia, 2003, pp. 135-147.

[5] B. Madan, K. Goseva-Popstojanova, K. Vaidyanathan, and K. Trivedi, "Modeling and Quantification of Security Attributes of Software Systems," *Int'l Conf. Dependable Systems and Networks*, 2002, pp. 505-514.

[6] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," *Proc. 6th Annual ACM/IEEE Mobile Computing and Networking*, Boston, Massachusetts, Aug. 2000, pp. 255-265.

[7] A. Mishra, K. Nadkarni, and A. Patcha, "Intrusion Detection in Wireless Ad-hoc Networks," *IEEE Wireless Communications,* vol. 11, no. 1, Feb. 2004, pp.48-60.

[8] D.M. Nicol, W.H. Sanders, and K.S. Trivedi, "Model-Based Evaluation: From Dependability to Security," *IEEE Transactions on Dependability and Secure Computing*, vol. 1, no.1, Jan.-Mar. 2004.

[9] A. Perrig and J.D. Tygar, *Secure Broadcast Communication in Wired and Wireless Networks*, Kluwer Academic Publishers, 2002.

[10] M. Steiner, G. Tsudik, and M. Waidner, "Diffie-Hellman Key Distribution Extended to Group Communication," *3rd ACM Conf. Computer and Communications Security*, New Delhi, 1996, pp. 31-37.

[11] S. Phoha, "Mission-oriented Sensor Networks," *IEEE Trans. on Mobile Computing,* vol. 3, no. 3, 2004, pp. 209-210.

[12] D. Sterne, et al., "A General Cooperative Intrusion Detection Architecture for MANETs," *Proc. 3rd IEEE Int'l Workshop on Information Assurance*, Mar. 2005, pp. 57-70.

[13] D. Subhadrabandhu, S. Sarkar, F. Anjum, "A Framework for Misuse Detection in Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, 2006, pp. 274-304.

[14] B. Sun, K. Wu, and U. W. Pooch, "Alert Aggregation in Mobile Ad Hoc Networks," *Proc. 2003 ACM Workshop on Wireless Security*, ACM Press, San Diego, Sep. 2003, pp. 69-78.

[15] Y. Zhang, W. Lee, and Y.A. Huang, "Intrusion Detection Techniques for Mobile Wireless Networks," *Wireless Networks,* vol. 9, no. 5, Kluwer Academic Publishers, Sep. 2003, pp. 545-556.